# Automatic Building Detection from Aerial Images for Mobile Robot Mapping

Martin Persson, Mats Sandvall, and Tom Duckett
*Centre for Applied Autonomous Sensor Systems*
*Örebro University, SE-701 82 Örebro, Sweden*
{*martin.persson, tom.duckett*}*@tech.oru.se, mats@sandvall.com*

***Abstract*— To improve mobile robot outdoor mapping, information about the shape and location of buildings is of interest. This paper describes a system for automatic detection of buildings in aerial images taken from a nadir view. The system builds two types of independent hypotheses based on the image contents. A segmentation process implemented as an ensemble of SOMs (Self Organizing Maps) is trained and used to create a segmented image showing different types of roofs, vegetation and sea. A second type of hypotheses is based on an edge image produced from the aerial photo. A line extraction process uses the edge image as input and extracts lines from it. From these edges, corners and rectangles that represent buildings are constructed. A classification process uses the information from both hypotheses to determine whether the rectangles are buildings, unsure buildings or unknown objects.**

***Index Terms*— Automatic building detection, aerial images, semi-autonomous mapping**

## I. Introduction

A mobile robot is an unmanned vehicle that can be autonomous, semi-autonomous or teleoperated. For an autonomous robot to be able to navigate, it needs a representation of the environment, a map. In order to fulfil assignments in unknown environments, a lot of research has focused on map building using the robot's on-board sensors. Most research has been devoted to indoor robots that have a well-structured environments containing flat floors, straight walls, etc. Outdoor map building introduces new problems. We cannot any longer assume that the ground is flat, there are larger moving objects such as cars, and the operating area has a larger scale that put higher demands on the localization algorithms.

Our research project focuses on a method to improve mapping of urban environments using a priori knowledge and information, called semi-autonomous mapping (SAM). The idea is to use pre-known information to enhance the mapping process, and in this case the information comes from an aerial image. This paper presents a new method for automatic building detection that will later be used in the mapping process.

Extraction of man-made structures from aerial images is difficult for many reasons. Aerial images have a wide variety of structured and unstructured content. They have different properties that make it hard to develop generic algorithms and methods for the extraction. Monocular 2D images restrict the possibilities for detection and classification. Images differ in scale (resolution), sensor type, orientation, quality, dynamic range, light conditions, different weather and seasons, etc. Buildings may have complicated structures and can be occluded by other buildings or vegetation. Together this gives a challenging research problem.

The new system presented in this paper is used to process an aerial photo. The aerial photo is a color photograph taken from the air, for example by an UAV, from a nadir view. The system is first trained with a set of training data for the segmentation process. The training data is acquired from an aerial photograph taken with the same camera as the aerial photos. The system builds two independent hypotheses concerning the contents of the image, which are then combined to give the final result.

The outline of the paper is as follows. Section II presents related work. In Section III the first type of hypothesis is described. In a segmentation process an ensemble of SOMs (Self Organizing Maps) is used to create a segmented image. The second type of hypothesis is described in Section IV. Here an edge image is produced from the aerial photo. A line extraction process uses the edge image as input and extracts lines from it. From the lines corners are constructed. The corners are then paired to create rectangles that represent building hypotheses. Finally, in Section V, a classification process uses the information from both hypotheses to classify whether the rectangles are buildings, unsure buildings or unknown objects.

## II. Related work

Detection of man-made structures, such as buildings, roads and vehicles, in aerial or satellite images has been an active research topic for many years. In aerial photogrammetry, extraction of objects is an important field. Aerial images, with their highly detailed contents, are an important source of information for applications including GIS, surveillance, etc. Modern sensor technology makes it possible to build accurate 3D-models using laser scanners [1]. SAR images, multi-spectral images, and high resolution satellite images give new possibilities in building detection [2], [3].

A survey [4] by Mayer focuses on extraction of buildings. Seven systems developed between 1984 and 1998 are assessed according to a number of criteria. The author has concentrated on models and strategies in this survey, but refers to other surveys that give an overview of the whole area of techniques for building extraction.

A basic system for detection of buildings may include i) edge detection, e.g., with the Canny algorithm, in a

greyscale image ii) rupture point detection. iii) line determination, and iv) search for buildings represented as orthopolygonal lines [5].

In the PhD thesis by C. Lin [6], generic 3D rectilinear models are used to model building parts. The system uses both wall and shadow information to verify hypotheses about modeled buildings. Edges are identified with a direction, where the direction is dependent on the brightness on the respective side of the edge. In this way parallel edges belonging to the same building can be connected. Shadows are detected by their darker appearance.

Cord et. al. presented a method for extraction and modelling of urban buildings [7]. By use of stereo images a digital elevation model (DEM) is created. The elevation information is then used for classification of buildings, ground surface and vegetation. The authors believe that altitude is one of the most important sources of information for building detection. This information is necessary to separate buildings from other man-made structures, e.g., parking lots.

Many other authors also use elevation information coming from, e.g., stereo images, radar, or laser scanning. In [8] elevation data is combined with a learning-based post-processing step that corrects positive false detections and can be used for finding negative false buildings. The authors use depth, colour and brightness, texture, and boundary energy in a tree classifier. The depth classifier filters out ground objects and lower vegetation. The color and texture classifier filters out vegetation that has the same height as the buildings. Finally a combined gradient field is used to determine the size and orientation of the buildings.

We are interested in methods that use a priori information, e.g., maps or GIS, in the extraction process. Carroll presented a system for change detection of features, which is used in an application for updating information about buildings in urban areas [9]. The presented prototype, HouseDiff, combines GIS and edge detection.

Another approach uses specific knowledge represented in digital topographic databases for improvement of automated image analysis for extraction of settlement areas [10]. Thus a model-driven top-down approach can be integrated into the commonly data-driven bottom-up process of satellite image analysis.

A third approach generates 3-D building hypotheses in dense urban areas using scanned maps and aerial images [11]. Maps are analysed in order to obtain a structural description of the scene. This information is then used for the analysis of a disparity image generated with a stereo pair of aerial images. According to the authors other approaches (not using maps) have been shown to perform well for sparse buildings, but not in dense urban areas.

Automatic detection of man-made structures is not yet a fully mature subject. Developed software is often limited to certain types of images giving a strong dependence on the input data for good performance. Elevation data, coming from, e.g., stereo images, is considered as very important in the detection process. A second point is that multiple view images gives different aspect angles which can help an automated system in the detection phase.

One problem that often seems to be ignored is the dynamics of aerial images; first of all variation in weather and seasons and secondly moving objects.

Many systems in this field use line and edge detection, form hypothesis, and try to fit those to 3D models for verification. It seems that color and texture are not so commonly used as features in the detection phase. Digital Surface Models, DSM, are however used by a number of authors.

## III. Hypotheses from Segmentation

The purpose of segmentation is to create an image that determines what every pixel represents, for example, vegetation, roofs and so on. The resulting image is called the segmented image that will be used later in the classification process.

The colour representation HSV (Hue, Saturation, Value) is used in the segmentation process in order to reduce sensitivity to intensity changes between images. In HSV space the colour (hue) is separated from the intensity, in contrast to RGB, where the intensity affects all three colour parameters, red, green, and blue. Initially a Bayes classifier was used for the segmentation, but after problems due to the discontinuity in the hue parameter were noticed, ESOM was used instead. (This discontinuity is due to the representation of the hue parameter as an angle between 0 and 360 degrees.)

### *A. Ensemble of Self Organising Maps (ESOM)*

Self-organising maps (SOM) or Kohonen maps were developed by Teuvo Kohonen [12] in the early eighties. The SOM is an unsupervised neural network that clusters similar data into similar categories. In an ensemble of SOMs, ESOM, every class is represented by a separate SOM network, see Figure 1. One benefit of using ESOM instead of other classification methods is that in the ESOM classes can be added or removed without the need to retrain the entire classifier.

The test vector is tested pixel-wise against every SOM-network. Each network returns a quantization-error ($q_{error}$) which is the distance between the sample vector and the best matching unit in the SOM. The SOM with the lowest $q_{error}$ is chosen as the winner. But even if the sample vector is not close to any of the SOM nets, one of the nets will be the closest one. To be useful in a classification system, the $q_{error}$ must be below a certain threshold, otherwise it is classified as “unknown”, i.e., it belongs to none of the known classes.

In order to ensure that all classes were equally represented in the trained classifier, all of the SOM networks were initialized with a fixed grid of 20x20 neurons.

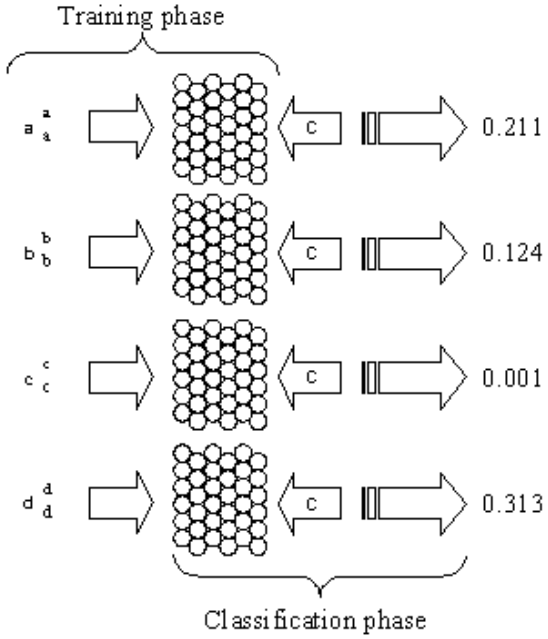


Fig. 1. Ensemble of SOMs (ESOM). A separate network is trained for each class in the training data. After training, the SOM with the unit that best matches the input vector is chosen as the winner.

### B. Training Data

The most interesting areas in the image are of course different types of roofs. Four types were trained in the system: red roofs, dark roofs, light roofs and copper roofs. Vegetation and sea are interesting as well, since it is good to know which areas are not roofs.

The training data that has been collected for this project is taken from the same series of images as the investigated areas. Experience showed that it is hard to separate light roofs and roads, so instead of trying to separate the two classes, they were merged into one class. The roads will later be removed in the classification step in Section V. Examples of used training data are shown in Figure 2.

Cameras have different spectra resulting in different shades of colours in the images, which may in turn affect the segmentation result. It is therefore recommended that the training set is taken from images coming from the same camera as the "real" aerial photos.

Fig. 2. Examples of training images (from left): Red roofs, Vegetation, Dark roofs, Light roofs / roads, Green roofs, and Sea.

## IV. Hypotheses from Line Extraction

The purpose of line extraction is to gain information about the various geometrical shapes in the image, i.e., the outline of the buildings. The complete process contains the following steps: (i) preparation of an edge image, (ii) line extraction, (iii) corner detection and (vi) merging lines to form rectangles.

### A. Binary Edge Image

An edge can be defined as a boundary between two regions with relatively distinct grey level properties. Basic edge detection is done by calculating the derivate of an image. There exist many different methods for this. We use a Canny edge detector [13] on a greyscale version of the input image. The output is a binary image with the same size as the input image.

### B. Line Extraction

For the line extraction two methods have been implemented. First, we used the Hough transform [14] that is widely used for line extraction. After implementing this two major drawbacks were found. First of all the Hough transform is widely known to be a time consuming algorithm. Secondly, problems with unwanted line merging appeared when using large windows. To get satisfying results the image had to be searched via small windows with a size of 10 by 10 pixels.

Instead an algorithm[1] developed by Peter Kovesi of the University of Western Australia was used. The algorithm is divided into three steps; (i) edge labeling, (ii) line segmenting and (iii) line merging. The purpose of the algorithm is to extract the lines from a binary edge image.

The edge labeling step takes an edge image as input. The algorithm looks for segments where all edge pixels are connected to another edge pixel. Two pixels are considered to be connected if they are eight-connected. Eight-connected means that in a binary image at least one of the surrounding eight pixels is in the same state (white or black) as the current pixel.

Each segment is labeled with a number from 1 to $n$ and stored in a vector of the form $[x_1, y_1; x_2, y_2; \ldots; x_m, y_m]$. The vector with all label segments is used as input by the line segmenting function.

The line segmenting algorithm takes one more argument as input, a maximum allowed deviation $D_{allow}$ from the original line. From the start point $(x_1, y_1)$ to the end point $(x_m, y_m)$ of the vector a virtual line is drawn. The maximum deviation $D_k$ from the original line is calculated, see Figure 3.

If the deviation is greater than $D_{allow}$, the line is cut in half at the point of the maximum deviation, creating a new vector, and continuing to work on the current line. The process is repeated with all vectors including the new one. If an original line is bifurcated the algorithm follows one of the branches and makes a new vector of the other. Using this algorithm all edge lines are broken down into smaller line segments that all meet the desired requirements.

The small lines are then merged together using the third part of the algorithm. This algorithm needs limits for

[1]Matlab functions are found at http://www.csse.uwa.edu.au/ pk/Research/MatlabFns/

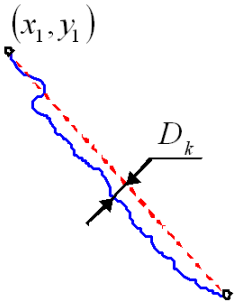


Fig. 3. Segmentation of a line.

the maximum tolerated angle deviation between two line segments that are about to be merged and the maximum distance between end points. If both values are within the limits the two original lines are merged. The new line is created using the old end points. When all merging is done, the length of the longest line is calculated and stored as $l_{long}$.

### C. Corner Detection

A corner detection algorithm is used to find corners from the line segments acquired in the line extraction process. A corner is defined as two lines with end points at a maximum distance of $l$ and a $90 \pm \beta_{dev}$ degree angle in between. Due to noise and the angle of the camera all corners may not be exactly 90 degrees, so the tolerance $\beta_{dev}$ was introduced. During the experiments $\beta_{dev} = 4°$ was used. When two lines fulfil the requirements the two lines are merged and the point where the two lines intersect is calculated and stored as a corner.

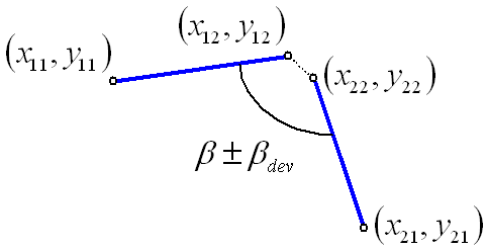


Fig. 4. Detection of corners.

### D. Rectangle Extraction

Since the buildings we are looking for are rectangular, it makes sense to form rectangles with the help of the corners. These are later classified with a scale ranging from definitely not a building to most probably a building.

From every centre point $(x_{corner}, y_{corner})$ in the corner an arc with radius $r$ defines the area within which a matching corner is searched for. Since $l_{long}$ is the longest line segment no house should be larger than that. $l_{long}$ can also be user defined if the user has knowledge of the sizes of buildings that are going to be identified. Corners

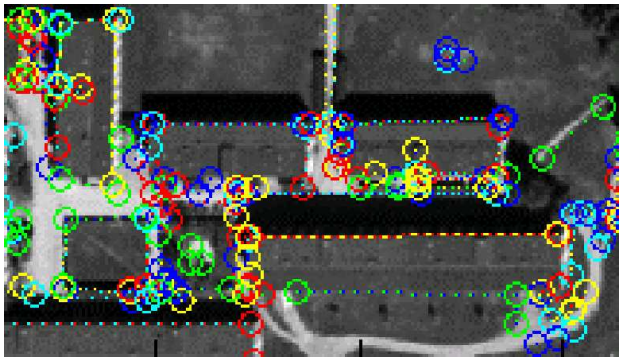

Fig. 5. Detected corners.

that are facing each other and have a maximum deviation of 10 degrees are merged to form a rectangle. Rectangles are formed without consideration of the image content. The classification is handled in the next step. When the rectangles are formed no consideration is taken of the length of the individual lines that forms the corners. If a rectangle is smaller than one of the lines the line is simply cut to fit in the rectangle. All corners can be used multiple times to form rectangles with different corners.

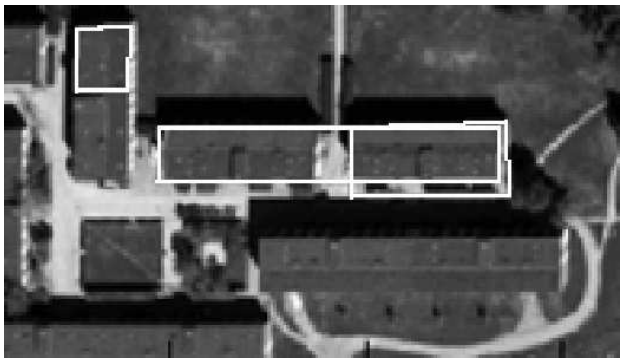

Fig. 6. Detected rectangle in image.

## V. Classification

In the classification step the hypotheses from line extraction are validated with those from the segmentation process. All rectangles are classified with a value between 0 and 1. 0 is definitely not a building and the higher the value is, the higher is the probability that the rectangle represents a building. To determine if a rectangle is a building, the area of the rectangle in the segmented image is examined.

Each segment's share of a rectangle is calculated as a percentage. The seven different segments are $\vartheta_{lror}$, $\vartheta_{drf}$, $\vartheta_{nat}$, $\vartheta_{sea}$, $\vartheta_{grf}$, $\vartheta_{rrf}$ and $\vartheta_{unk}$. To determine the probability of a house we deal with two cases. The first case is when the unknown percentage, $\vartheta_{unk}$, is larger then 25% and the second is when it is lower. The motivation for this is as follows. When a rectangle consists of more than a

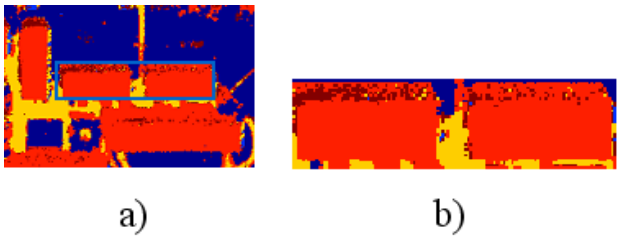


Fig. 7. a) A rectangle plotted over the segmented image. b) The contents of the rectangle.

quarter of unknown material, the amount of uncertainty is considered to be too large. These rectangles are therefore rejected. The rectangles that are not rejected are divided into two more cases. Case one is when the light roofs or roads percentage ($\vartheta_{lror}$) is the dominating segment (larger than 20%). The second is when $\vartheta_{lror}$ is below 20%. When $\vartheta_{lror}$ is larger than 20% the rectangle is treated as an uncertain building. This is because of the uncertainty in the interpretation of light areas. Often these areas are roads, but they can also be buildings. In the tested images an average of 22% of the correctly found building area were detected as uncertain buildings, and on average 62% of incorrectly found building area was detected as uncertain buildings.

The next step in the classification step is to add the two largest values from the following set $\{\vartheta_{lror}, \vartheta_{drf}, \vartheta_{grf}, \vartheta_{rrf}\}$ and if $\vartheta_{unk}$ is less than 25%, $\vartheta_{unk}$ is also added. The total sum is called the building rating and is denoted $\vartheta_{building}$. The purpose of taking the two largest values is simply that buildings can have several different roofs.

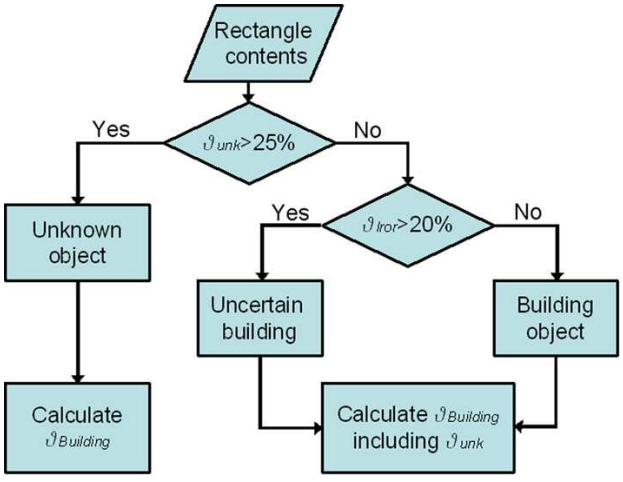


Fig. 8. Diagram of the classification process.

## VI. Evaluation

In order to evaluate the performance of the building detection the following measures were introduced: $\varphi_{corr}$, $\varphi_{wro}$, $\varphi_{nfd}$ are the percentage of correctly found areas, incorrectly found areas and not found area respectively. $\varphi_{fcor}$, $\varphi_{fwro}$ represent the share of found area that is classified correctly and incorrectly respectively. $\varphi_{nfd}$ is defined as the building area that is not covered by a rectangle and $\varphi_{corr}$ is defined as the building area that is covered by a rectangle. $\varphi_{wro}$ is defined as the area which an rectangle covers that is not a building. The values are transformed from area to percentage with the total building area used as the basis. Therefore some values can exceed 100%.

To measure the performance of the system, binary maps were handmade to pinpoint the actual location of the buildings. These were then used as a ground truth information for the evaluation.

The used aerial photo was taken at a height of 4 600 meters over Almby in Örebro, with a resolution of 0.25 meters per pixel. One part of it, the Campus area at Örebro University, see Fig. 9, was used for the examples presented in this paper. Other parts used in the evaluation include residential areas.

Fig. 9. The Campus area at Örebro University.

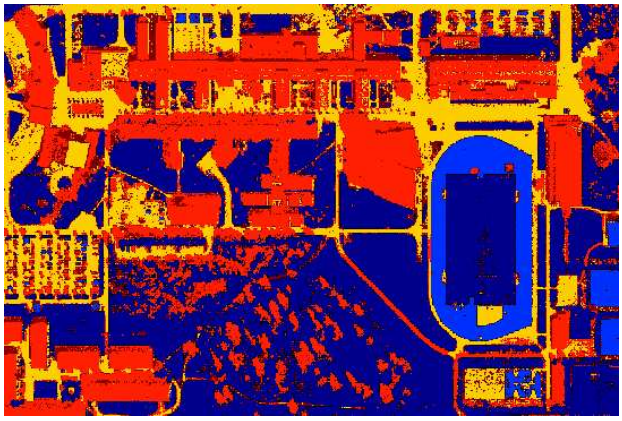

Fig. 10. The segmented image using ESOM. Light blue is red roofs, dark blue is nature, red is dark roofs, yellow is light roofs / roads and dark red is unknown.

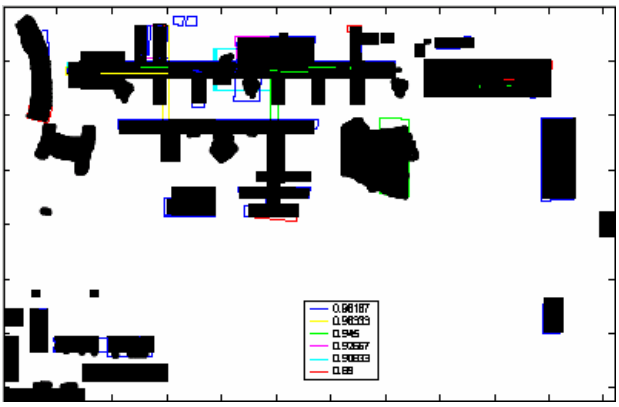

Fig. 11. Results without unsure buildings. With building ratios from 0.89 to 1. The black rectangles are the ground truth locations of the buildings.

The overall detection rate for the campus area is fairly good with 53% correctly found areas ($\varphi_{corr}$), of which 93% was classified correctly ($\varphi_{fcor}$). The unsure buildings were left out. The major problem is that the program connects the two largest building complexes in the image. One can also note that the system does not find any buildings that are close to the edges of the image due to the fact that no edges are found there. Introducing the uncertain buildings the results change, $\varphi_{corr}$ increases to 65 % and $\varphi_{fcor}$ decreases to 80 %.

The results including both Campus and the residential areas are presented in Table I. As seen in the table the discovered area is at 0.8 already quite close to 50% with a correctness close to 70%.

| | | $\varphi_{corr}$ | $\varphi_{wro}$ | $\varphi_{nfd}$ | $\varphi_{fcor}$ | $\varphi_{fwro}$ |
|---|---|---|---|---|---|---|
| 0.7 | Mean | 62% | 39% | 65% | 61% | 39% |
| | Max | 74% | 56% | 116% | 80% | 59% |
| | Min | 44% | 26% | 32% | 41% | 20% |
| 0.8 | Mean | 47% | 53% | 29% | 69% | 31% |
| | Max | 64% | 75% | 58% | 87% | 54% |
| | Min | 25% | 36% | 15% | 46% | 13% |

TABLE I

RESULTS FROM BOTH CAMPUS AND THE RESIDENTIAL AREAS WITH BUILDING RATINGS FROM 0.7 TO 1 AND 0.8 TO 1

## VII. CONCLUSION

The goal of this work was to make an implementation that can extract buildings from aerial images. The number of detected buildings is fairly high. Since the result is calculated with area as basis, the result is a bit misleading. The number of partially or totally found buildings in the Campus image is 14 out of 17 possible, which gives a percentage of correctness of 82% with zero false positives. All rectangles are in fact covering some part of a building.

To be able to truly distinguish buildings from other man-made objects, information about the elevation of the area in the image is needed. This could be obtained by, e.g., laser, radar, stereovision or by an outdoor mobile robot. This would result in an image that looks like the segmented image, but instead of pixel colour information, the elevation for every pixel would be available. All detected areas could then be classified using both colour and elevation.

The program has been evaluated with different types of images, and the conclusion is that it works best with images that contain buildings and vegetation. The hardest images are pictures of an inner city, mainly due to the vast amount of light coloured roads. No preprocessing, e.g. edge enhancing, has been performed outside the program.

The system has given us a platform for continued research in outdoor mapping with a mobile robot. The next step in our work is to connect the output from this system to the navigation sensors on board the robot.

## REFERENCES


[1] Ulf Söderman, Simon Ahlberg, Åsa Persson, and Magnus Elmqvist. Towards rapid 3D modelling of urban areas. In *Proceeding of the Second Swedish-American Workshop on Modeling and Simulation, (SAWMAS-2004)*, Feb 2004.

[2] F. Tupin and M. Roux. Detection of building outlines based on the fusion of SAR and optical features. *ISPRS Journal of Photogrammetry & Remote Sensing*, 58:71–82, 2003.

[3] R. Xiao, C. Lesher, and B. Wilson. Building detection and localization using a fusion of interferometric synthetic aperture radar and multispectral image. In *Proc. ARPA Image Understanding Workshop*, pages 583–588, 1998.

[4] H. Mayer. Automatic object extraction from aerial imagery – a survey focusing on buildings. *Computer vision and image understanding*, 74(2):138–149, May 1999.

[5] Luiz Alberto Cardoso. Computer aided recognition of man-made structures in aerial photographs. Master's thesis, Naval postgraduate school, Monterey, California, 1999.

[6] Chung-An Lin. *Perception of 3-D Objects from an Intensity Image Using Simple Geometric Models.* PhD thesis, Faculty of the Graduate School, University of Southern California, Dec 1996.

[7] M. Cord, M. Jordan, JP. Cocquerez, and N. Paparoditis. Automatic extraction and modelling of urban buildings from high resolution aerial images. In *IAPRS 1999*, Sep 1999.

[8] Yanlin Guo, Harpreet Sawhney, Rakesh Kumar, and Steve Hsu. Learning-based building outline detection from multiple aerial images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–545–52, 2001.

[9] Robert W. Carroll. Detecting building changes through imagery and automatic feature processing. In *URISA 2002 GIS & CAMA Conference Proceedings*, 2002.

[10] Klaus-Jürgen Schilling and Thomas Vögtle. An approach for the extraction of settlement areas. In A. Gruen and H. Li, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, pages 333–342. Birkhäuser Verlag, 1997.

[11] Michel Roux and Henri Maître. Three-dimensional description of dense urban areas using maps and aerial images. In A. Gruen and H. Li, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, pages 311–322. Birkhäuser Verlag, 1997.

[12] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[13] John Canny. A computational approach for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):279–98, Nov 1986.

[14] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing*. Prentice-Hall, 2002.